5x

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/931,649	08/15/2001	Debi Mishra	MS1-929US	2836

22801 7590 09/14/2004

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
INGBERG, TODD D	
ART UNIT	PAPER NUMBER
2124	

DATE MAILED: 09/14/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

## *Office Action Summary*	**Application No.** 09/931,649	**Applicant(s)** MISHRA ET AL.	
	Examiner Todd Ingberg	**Art Unit** 2124	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 April 2002*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-31* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-31* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *15 August 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: _____.

DETAILED ACTION

Claims 1 – 31 have been examined.

Oath/Declaration

1. The addition of two inventors has been noted. Request to update the Bibliographic Data Sheet has been internally submitted within the USPTO. No further action is required by Applicant.

Claim Rejections - 35 USC § 101

2. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

3. Claims 1 – 8, 10 –11, 13-14, 16 – 18 , 20 – 24, 26 – 28 and 30 are rejected under 35 U.S.C. 101 for failing the concrete and tangible tests. Examiner has shown a simple way to overcome this rejection below.

Claim 1
A method **executing on a computer readable medium** comprising: receiving an initial code associated with a first framework, the first
framework having an object hierarchy; and converting the initial code to a converted code that combines the object hierarchy of the first framework with an object hierarchy of a second framework.

Claim 10
A method **executing on a computer readable medium** comprising: receiving an initial code associated with a first framework, the first
framework having an exception hierarchy; and converting the initial code to a converted code that combines the exception hierarchy of the first framework with an exception hierarchy of a second framework.

Claim 13
A method **executing on a computer readable medium** comprising: receiving an initial code associated with a first framework, the first
framework having an exception hierarchy; and converting the initial code to a converted code that maps the exception I hierarchy of the first framework to an exception hierarchy of a second framework.

Claim 16
A method **executing on a computer readable medium** comprising: receiving an initial code associated with a first framework, the first
framework having reflection transparency; and converting the initial code to a converted code that supports the reflection transparency of the first framework on a second framework.

Claim 20
A method **executing on a computer readable medium** comprising: receiving an initial code associated with a first framework, the first
framework having scoping; and converting the initial code to a converted code that supports the scoping of the first framework on a second framework.

Claim 26
A method **executing on a computer readable medium** comprising: receiving an initial code associated with a first framework, the first
framework having type characteristics; and converting the initial code to a converted code that supports the type characteristics of the first framework on a second framework.

Claim 30
A method **executing on a computer readable medium** comprising: receiving an initial code associated with a first framework, the first
framework having at least one member selected from the group consisting of object hierarchies, exception hierarchies, type characteristics, reflection transparencies, and scoping; and converting the initial code to a converted code that supports at least one of the selected members on a second framework.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1 – 4, 7 – 9, 16 – 20, 22, 25 – 26, 29 and 30 are rejected under 35 U.S.C. 102(b) as being anticipated by "Flexible Collaboration Transparency Supporting Worker Independence in Replicated Application-Sharing Systems", by James **Begole** et al, ACM June 1999.

Claim Interpretation

The following is the patentable weight and interpretation given to the following limitations:

A. a **superclass** named java.lang.Object – the name java.lang.Object is viewed as data and not given patentable weight. The parent or superclass is given patentable weight.
B. an **array class** named System.Array – the name SystemArray is viewed as data and not given patentable weight. The use of the container class of an array is given patentable weight.

Claim 1

Begole anticipates a method comprising: receiving an initial code associated with a first framework, the first framework having an object hierarchy; and converting the initial code to a converted code that combines the object hierarchy of the first framework with an object hierarchy of a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122.

Claim 2

The method of claim 1 wherein the converting produces a class that inherits from a class of the first framework.

Examiner's Response

Figures 4 on page 109 show the underlying object models of the transformation.

Claim 3

The method of claim 2 wherein the class of the first framework comprises a superclass of the first framework.

Examiner's Response

Figures 4 on page 109 show the underlying object models of the transformation.
Where Panel is a superclass.

Claim 4

The method of claim 2 wherein the class of the first framework comprises a superclass named java.lang.Object.

Examiner's Response

As per claim 3. Data not given patentable weight.

Claim 7

The method of claim 1 wherein the converting includes creating a new class.

Examiner's Response

Figures 4 on page 109 show the underlying object models of the transformation.

Claim 8

The method of claim 7 wherein the new class inherits from java.lang.Object and from System.Array.

Examiner's Response

As per claim 5 and data not given patentable weight.

Claim 9

A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having an object hierarchy, to a converted code that combines the object hierarchy of the first framework with an object hierarchy of a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122.

Claim 16

A method comprising: receiving an initial code associated with a first framework, the first framework having reflection transparency; and converting the initial code to a converted code that supports the reflection transparency of the first framework on a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 109 show the end user display and underlying object models of the transformation. Reflection transparency is anticipated by Collaboration Transparency as per page 96 first and second paragraphs.

Claim 17

The method of claim 16 wherein the converting includes checking for methods associated with the reflection transparency of the first framework.

Examiner's Response

As per claim 16

Claim 18

The method of claim 16 wherein the converting includes rendering a stack entry transparent.

Examiner's Response

Inherent in Collaboration Transparency as per page 96 first and second paragraphs.

Claim 19

A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having reflection transparency, to a converted code that supports the reflection transparency of the first framework on a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122.

Claim 20

A method comprising: receiving an initial code associated with a first framework, the first framework having scoping; and converting the initial code to a converted code that supports the scoping of the first framework on a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122.

Claim 22

The method of claim 20 wherein the converting includes marking a package scope associated with the first framework as an assembly on the second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122. Figure 4 shows class to class copy maintaining scope. Page 98 mentions private and shared which involve scope – last paragraph of 2.1

Claim 25

A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having scoping, to a converted code that supports the scoping of the first framework on a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122. Also, details pages 110 to 122. Figure 4 shows class to class copy maintaining scope. Page 98 mentions private and shared which involve scope – last paragraph of 2.1

Claim 26

A method comprising: receiving an initial code associated with a first framework, the first framework having type characteristics; and converting the initial code to a converted code that supports the type characteristics of the first framework on a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122. Also, details pages 110 to 122. Figure 4 shows class to class copy maintaining scope. Page 98 mentions private and shared which involve scope – last paragraph of 2.1

Claim 29

A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having type characteristics, to a converted code that supports the type characteristics of the first framework on a second framework.

Examiner's Response

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122. Also, details pages 110 to 122. Figure 4 shows class to class copy maintaining scope. Page 98 mentions private and shared which involve scope – last paragraph of 2.1

Claim Rejections - 35 USC § 103

6. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

7. Claims 5 – 6 and 8 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Flexible Collaboration Transparency Supporting Worker Independence in Replicated Application-Sharing Systems", by James **Begole** et al, ACM June 1999 in view of standard containers such as an array as taught by **C++,** B. Stroustrup in 1997.

Claim 5
The method of claim 2 wherein the class of the second framework comprises an array class.
<u>Examiner's Response</u>
Although, Begole teaches conversion of one structure to another as per claim 1. Begole does not explicitly teach the underlying structure using a container class. It is C++ who teaches using standard container classes and specifically an array (C++, page 496, array), therefore, it would have been obvious to one of ordinary skill in the art at the time of invention to combine Begole and C++ because container classes provide for storage.

Claim 6
The method of claim 2 wherein the class of the second framework comprises an array class named System.Array.
<u>Examiner's Response</u>
As per claim 5. Data not given patentable weight.

Claim 8 (rejected twice)
The method of claim 7 wherein the new class inherits from java.lang.Object and from System.Array.
<u>Examiner's Response</u>
As per claim 5 and data not given patentable weight.

8. Claims 10 – 15 and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Flexible Collaboration Transparency Supporting Worker Independence in Replicated

Application-Sharing Systems", by James **Begole** et al, ACM June 1999 in view of Exception classes as taught by **C++,** B. Stroustrup in 1997.

Examiner's Response to Claims 10-15, 30 and 31

Begole teaches in Figures 3 and 4 on pages 108 – 109 show the end user display and underlying object models of the transformation. Reflection transparency is anticipated by Collaboration Transparency as per page 96 first and second paragraphs. Despite, Begole teaching on pages 111 to 113 teaches JAMM (JAVA Applets Made Multi-User). Begole does not disclose the details an exception hierarchy for JAVA. C++ on page 385 teaches an exception hierarchy. It would have been obvious to one of ordinary skill in the art to combine Begole's teaching of JAMM with the knowledge of exception hierarchy of C++ , because understanding a language exception hierarchy provides for Collaboration Transparency (not visible by the user).

Claim 10

A method comprising: receiving an initial code associated with a first framework, the first framework having an exception hierarchy; and converting the initial code to a converted code that combines the exception hierarchy of the first framework with an exception hierarchy of a second framework.

Claim 11

The method of claim 10 wherein the converting includes mapping exceptions.

Claim 12

A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having an exception hierarchy, to a converted code that combines the exception hierarchy of the first framework with an exception hierarchy of a second framework.

Claim 13

A method comprising: receiving an initial code associated with a first framework, the first framework having an exception hierarchy; and converting the initial code to a converted code that maps the exception hierarchy of the first framework to an exception hierarchy of a second framework.

Claim 14

The method of claim 13 wherein the converting includes combining exception hierarchies.

Claim 15

A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having an exception hierarchy, to a converted code that maps the exception hierarchy of the first framework with an exception hierarchy of a second framework.

Claim 30
A method comprising: receiving an initial code associated with a first framework, the first framework having at least one member selected from the group consisting of object hierarchies, exception hierarchies, type characteristics, reflection transparencies, and scoping; and converting the initial code to a converted code that supports at least one of the selected members on a second framework.

Claim 31
A computer-readable medium storing computer-executable instructions to convert an initial code associated with a first framework, the first framework having at least one member selected from the group consisting of object hierarchies, exception hierarchies, type characteristics, reflection transparencies, and scoping, to a converted code that supports at least one of the selected members of the first framework on a second framework.

9. Claims 27 – 28 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Flexible Collaboration Transparency Supporting Worker Independence in Replicated Application-Sharing Systems", by James **Begole** et al, ACM June 1999 unanticipated sharing (Begole page 105 in view of casting as taught by **C++,** B. Stroustrup in 1997.

__Examiner's Response to Claims 27 and 28__
Begole teaches preparing for unanticipated sharing (Begole page 105), but does not disclose the details on how to handle unexpected sharing. Having defaults for undefined types and converting them into a known type using a casting operation is well within the knowledge of one of ordinary skill in the art. C++ teaches casting operations (C++. page 130-131) . therefore, it would have been obvious to one of ordinary skill in the art to combine the teaching of Begole with C++ because providing for a default type when an undefined type is encountered provides support for unexpected sharing.

Claim 27
The method of claim 26 wherein the converting supports type characteristics of the first framework related to casting between real and integer types on the second framework.

Claim 28
The method of claim 26 wherein the converting supports type characteristics of the first framework related to overflow and undefined types on the second framework.

10. Claims 21, 23 and 24 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Flexible Collaboration Transparency Supporting Worker Independence in Replicated

Application-Sharing Systems", by James **Begole** et al, ACM June 1999 in view of Access as taught by **C++,** B. Stroustrup in 1997.

Examiner's Response for Claims 21 , 23 and 24

Figures 3 and 4 on pages 108 – 110 show the end user display and underlying object models of the transformation. Also, details pages 110 to 122. Also, details pages 110 to 122. Figure 4 shows class to class copy maintaining scope. Page 98 mentions private and shared which involve scope – last paragraph of 2.1. Although Begole teaches converting and maintaining the scope, and Begole teaches unanticipated sharing (Begole page 105). The exact mechanisms to handle unexpected sharing are not disclosed.

Scope of PRIVATE, PUBLIC OR PROTECTED is inherent in object oriented programming languages such as C++. PUBLIC access is taught in C++ (C++ page2 850. When selecting a default one of ordinary skill in the art would select PUBLIC because it is old and well known to one of ordinary skill in the art that PUBLIC allows all callers to access the methods and attributes of scope PUBLIC. Therefore, it would have been obvious for one of ordinary skill in the art at the time of invention to select PUBLIC of the three choices because, the functionality of the code can be tested prior to determining the actual desired scope and altering the code (**Note**: after this scope modification the behavior not the functionality would be the focus of testing).

Motivation Specific to Invention

Also, at this point the method calls and their relationships are already defined by the software being converted. No behavioral impact would result.

Claim 21

The method of claim 20 wherein the converting includes marking a package scope and a protected scope associated with the first framework as a public scope on the second framework.

Claim 23

The method of claim 20 wherein the converting includes marking a protected scope associated with the first framework as an assembly or a family on the second framework.

Claim 24

The method of claim 20 wherein the converting includes marking, the marking selected from a member of the group consisting of marking a protected scope associated with the first framework as an assembly or a family on the second framework; marking a package scope associated with the first framework as an assembly on the second framework; marking a package scope and a protected scope associated with the first framework as a public scope on the second framework; and combinations thereof.

Correspondence Information

11. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Todd Ingberg whose telephone number is (703) 305-9775. The examiner can normally be reached during the following hours:

Monday	Tuesday	Wednesday	Thursday	Friday
6:15 – 1:30	6:15- 3:45	6:15 – 4:45	6:15-3:45	6:15-130

This schedule began December 1, 2003 and is subject to change.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kakali Chaki can be reached on (703) 305-9662. Please, note that as of August 4, 2003 the FAX number changed for the organization where this application or proceeding is assigned is (703) 872-9306.

Also, be advised the United States Patent Office new address is

Post Office Box 1450
Alexandria, Virginia 22313-1450

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-9700.

Special Notice

12. Please, Note the Examiner's telephone number will change in October when the Art Unit moves to the new location. The Examiner's new telephone number will be as follows:

(571) 272-3723

Todd Ingberg
Primary Examiner
Art Unit 2124
September 4, 2004